RS 译码优化及卫星通信 CCSDS 下 RS 译码仿真 *

王兵锐 [1, 2]，杨晓非 [2]，姚行中 [3+]

(1. 南阳师范学院 软件学院，河南 南阳 473007; 2. 华中科技大学 光学与电子信息学院，武汉 430074; 3. 火箭军研究院，北京 1000853)

摘　要：针对里德所罗门(RS)译码的关键步骤错误值求解不灵活的问题提出一种更加通用的求解算法。该算法融入多种本原元运算，使得对不同参数都普遍适用；针对该算法在求解错误值多项式时计算量过大，根据伽罗华域的特征提出了一种优化方法，从而省去一半运算以及节省存储资源。针对 RS 译码另一个步骤求解错误位置多项式时迭代复杂度过高的问题，经过对补偿差值的详细分析，给出了一种快速搜索迭代次数的算法，且迭代复杂度由 $O(n^2)$ 下降了一个数量级到 $O(n)$。以卫星通信中的国际空间数据系统咨询委员会(CCSDS)标准下 RS(255，223)为具体研究对象，结合优化后的译码算法进行了数据仿真分析和误比特率测试。实验结果表明，采用改进的求错误值算法和优化的迭代次数搜索算法，可以有效快速解码。

关键词：里德所罗门码；伽罗华域；错误值多项式

中图分类号：TN911.2　　doi: 10.19734/j.issn.1001-3695.2018.11.0839

Improved RS decoding algorithm and RS decoding simulation under CCSDS standard in satellite communications

Wang Bingrui[1, 2], Yang Xiaofei[2], Yao Xingzhong[3+]

(1. *School of Software*, *Nanyang Normal University*, *Nanyang Henan* 473007, *China*; 2. *School of Optics & Electronics*, *Huazhong University of Science & Technology*, *Wuhan* 430074, *China*; 3. *Rocket Army Academy*, *Beijing* 100085, *China*)

Abstract: Focusing on the inflexibility of error value solving in the key steps of RS decoding, this paper proposed a general algorithm. The algorithm dealt with a variety of primitive elements, making it universally applicable to different parameters. Then, with regard to large computational complexity of error value polynomial involved in the algorithm, this paper described an optimization method to eliminate half of the operation and save storage resources, according to characteristics of the Galois field. There is a problem that the iterative complexity is high when solving the error location polynomial of RS decoding. Based on the detailed analysis of compensation difference, an algorithm for fast search iterations was presented, and the complexity of search iteration was reduced from $O(n^2)$ to $O(n)$. Taking RS(255, 223) code under the satellite communication CCSDS standard as the specific object, combined with the proposed RS decoding algorithm, the paper carried out specific data simulation and BER test. Simulation experiments show that the improved error value solving algorithm and optimized fast search algorithm are effective.

Key words: reed solomon (rs) code; galois field; error value polynomial

0　引言

里德所罗门 (Reed Solomon, RS)码能纠正随机错误和突发错误，而且纠正后者能力更强。RS 码被广泛应用于数据存储和通信中。随着物联网技术的发展，越来越多的设备被连接并产生大量数据。通过将计算密集型任务转移到边缘设备，基于云的存储技术已成为主流。Wang 等人[1]提出了一种基于雾计算的数据同步新架构。并采用 RS 码来保证安全。Demirkan 和 Silvus[2]结合软输出维特比算法，提出了一种多重交织 RS 码架构，应用在垂直磁记录上。Li 等人[3]针对伴有加性高斯白噪声的突发瑞利衰落信道，提出了两种 RS 解码方法。在仅包含加性高斯白噪声的区域中，只需进行纠错处理。在突发衰落区域中进行纠错纠删处理。利用分子作为信息载体的纳米级通信中，分子通信是一种新的范例。在基于扩散的分子通信中，系统性能容易受连续分子交叉引起的符号间干扰的影响。为了解决这个问题，Dissanayake 等人[4]采用 RS 码来提高传输的可靠性。在基于单模光纤的直接检测光学系统，Chen 等人[5]从实验上阐述了 RS 码结合多符号交织实时正交频分复用信号传输。这是首次在实时光正交频分复用系统中采用 RS 码。Luby transform 和 Raptor 这些纠删码可以对计算机网络进行弹性纠删分布。Borujeny 和 Ardakani[6]提出了一类基于 RS 码的纠删码，能保证接收零开销。

RS 码的应用非常广泛。针对 RS 译码算法中两个关键步骤—求解错误位置多项式和求解错误值，本文进行了研究分析和优化。在求解错误位置多项式时，需要进行搜索迭代运算，其复杂度为 $O(n^2)$。当校验位和码长数值不断增大时，复杂度较高。本文提出一种改进方法，降低搜索复杂度是十分

收稿日期：2018-11-13；**修回日期：**2019-01-11　**基金项目：**军委科技委创新特区课题（17-H863-01-ZT-002-009-02）；河南省高等学校重点科研项目（18A520044）；南阳师范学院基金资助项目（501-17323）

作者简介：王兵锐（1986-），男，河南南阳人，讲师，博士，主要研究方向为纠错码、卫星通信；杨晓非（1963-），男，湖北武人，教授，博导，博士，主要研究方向为信号处理、智能传感器；姚行中（1962-），男（通信作者），湖北武人，研究员，博导，博士，主要研究方向为卫星图像处理、精确制导（study_research666@126.com）.

有意义的。在具体设计 RS 码时，会引入很多参数，使得参与计算的系数偏小且具有对称性。小系数更方便计算，系数对称可以节约存储。例如 32 个数据中前 16 个数据和后 16 个数据是对称的，那么只需存储一半数据。RS 码参数设计灵活，需要一个通用的求解错误值算法，但文献[7]介绍的求解错误值算法涉及的参数比较单一。本文给出一种更加通用的求解错误值算法具有一定的意义。在用于卫星通信的国际空间数据系统咨询委员会(Consultative Committee for Space Data Systems, CCSDS)标准下，依据改进的算法对 RS(255,223)进行了具体数值分析和误比特率仿真测试。

1 求解伴随式

对不超过 RS 码纠错能力的一组错误信息，RS 译码算法都可以检测其中的错误，确定错误位置多项式，找到错误发生的位置以及求出错误值并最终纠正过来。译码流程图如图 1 所示。

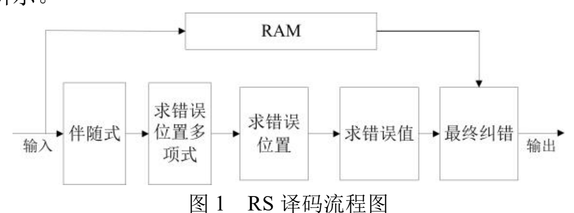


图 1 RS 译码流程图

Fig. 1 RS decoding flow chart

为方便研究，定义以下多项式。信息多项式可以表示为

$$m(x)=m_0+m_1x+\cdots+m_{k-1}x^{k-1} \tag{1}$$

编码后的多项式为

$$c(x)=c_0+c_1x+\cdots+c_{n-1}x^{n-1} \tag{2}$$

信道中产生的错误多项式为

$$e(x)=e_0+e_1x+\cdots+e_{n-1}x^{n-1} \tag{3}$$

接收到的多项式可以写为

$$r(x)=r_0+r_1x+\cdots+r_{n-1}x^{n-1} \tag{4}$$

多项式(2)(3)与(4)关系是 $r(x)=e(x)+c(x)$。含有 t 个错误的多项式表达式为

$$e(x)=e_{j1}x^{j1}+e_{j2}x^{j2}+\cdots+e_{jt}x^{jt} \tag{5}$$

其中：e_{j1}，e_{j2}，…，e_{jt} 为错误值，对应的 $j1$，…，jt 为错误位置。jt 作为一个整体代表错误位置，并不是 j 和 t 的乘积。为表达方便，用 Y_i 来代替 e_{ji}，X_i 代替 x^{ji}，$1\le i\le t$。式(5)转换为

$$e(x)=Y_1X_1+Y_2X_2+\cdots+Y_tX_t \tag{6}$$

生成多项式定义为

$$g(x)=\prod_{j=1}^{2t}(x-\alpha^{C*(b+j)}) \tag{7}$$

其中：C 和 b 为一个常数，α 为伽罗华域中的一个本原元，针对 CCSDS 标准，C 的取值为 11，b 的取值为 111。容易发现 $\alpha^{C*(b+j)}$ 是式(7)的根。把 $\alpha^{C*(b+j)}$ 代入到 $r(x)$ 中得到的值称为伴随式并记做 s。

$$s_j=r(\alpha^{C*(b+j)})=r_0+r_1(\alpha^{C*(b+j)})+\cdots+r_{n-1}(\alpha^{C*(b+j)})^{n-1} \tag{8}$$

其中：$1\le j\le 2t$，观察式(8)容易发现，伴随式可以采用循环迭代的方法实现。

$$s_j=(\cdots(r_{n-1}(\alpha^{C*(b+j)})+r_{n-2})(\alpha^{C*(b+j)})+\cdots+r_1\cdots)(\alpha^{C*(b+j)})+r_0 \tag{9}$$

2 求解错误位置多项式

2.1 原求解算法

根据 RS 码的编码理论[8]有

$$s_j=e(\alpha^{C*(b+j)})=\sum_{i=1}^{t}Y_i(\alpha^{C*(b+k)})^{ji} \tag{10}$$

RS 解码就是要确定错误值 Y_1 到 Y_t 和错误位置 $j1$ 到 jt。但难以直接求解，一般采用间接求解。定义错误位置多项式为

$$\sigma(x)=\prod_{i=1}^{t}(1-\alpha^{C*ji}x)=1+\sigma_1x+\cdots+\sigma_tx^t \tag{11}$$

一般采用 BM(Berlekamp-Massey)算法来求解错误位置多项[9,10]。设 μ 是迭代次数，$\sigma^{(\mu+1)}(x)$ 是第 μ 次迭代对应的错误多项式，d_μ 是第 μ 次的差值，l_μ 是 $\sigma^{(\mu)}(x)$ 的次数。为求解下一个多项式 $\sigma^{(\mu+1)}(x)$，必须验证差值 d_μ。有以下表达式：

$$d_\mu=s_{\mu+1}+\sigma_1^{(\mu)}s_\mu+\cdots+\sigma_{l_\mu}^{(\mu)}s_{\mu-l_\mu+1} \tag{12}$$

如果 d_μ 为 0，则有 $\sigma^{(\mu+1)}(x)=\sigma^{(\mu)}(x)$ 和 $l_{\mu+1}=l_\mu$。如果 d_μ 不为 0，则校正 $\sigma^{(\mu)}(x)$ 来求 $\sigma^{(\mu+1)}(x)$。校正方法：寻找在第 μ 次迭代之前的某次迭代 ρ，使 $d_\rho\ne 0$ 且 $\rho-l_\rho$ 最大，l_ρ 为 $\sigma^{(\rho)}(x)$ 的次数。则有

$$\sigma^{(\mu+1)}(x)=\sigma^{(\mu)}(x)-d_\mu d_\rho^{-1}x^{\mu-\rho}\sigma^{(\rho)}(x) \tag{13}$$

从第一次按照这个步骤操作直到 $2t$ 次迭代结束。那么寻找 ρ 次迭代是 BM 算法的一个关键步骤。

2.2 改进的搜索算法

对于式(13)，先假设 $\sigma^{(\mu)}(x)$ 和 $\sigma^{(\rho)}(x)$ 的次数都为 l_μ。当所有的 d_μ 不为 0 时，那么 d_ρ^{-1} 也不为 0。$\rho\sigma^{(\mu+1)}(x)$ 是 μ 次之前的某次迭代，所以 $\mu-\rho$ 大于 1，于是 $d_\mu d_\rho^{-1}x^{\mu-\rho}\sigma^{(\rho)}(x)$ 这一项的次数大于 l_μ，这样导致 $\sigma^{(\mu+1)}(x)$ 的次数大于 l_μ，这与刚才的假设矛盾。同时注意到，$\sigma^{(\rho)}(x)$ 是第 μ 次之前的任意一次，$\sigma^{(\rho)}(x)$ 可以是 $\sigma^{(\mu-1)}(x)$，也就是第 ρ 次可以是 $\mu-1$ 次。$\sigma^{(\mu+1)}(x)$、$\sigma^{(\mu)}(x)$ 和 $\sigma^{(\mu-1)}(x)$ 的次数都为 l_μ 这是不成立的。于是得出这样的结论。当 d_μ 从不为 0 时，任意连续 3 次的迭代，得到的多项式的次数不会相等。

又因 $\sigma(x)$ 的系数有 t 个，需要迭代 $2t$ 次才能求出 t 个系数[11]。l_ρ 的最大值为 t，$\sigma(x)$ 的次数 l_ρ 不能在 $2t$ 次迭代中呈每次加 1 的方式增加，那样就会达到 $2t$ 次。也就是说，任意连续的 3 次迭代，这 3 次产生的 $\sigma(x)$ 的次数不会是差值为 1 的 3 个等差数列。而且次数只会增加，不会减小。

根据之前的分析，当所有的 d_μ 不为 0 时，任意连续 3 次求出的 $\sigma(x)$ 的次数不会相等，也不会全部不相等。所以，$\sigma(x)$ 的次数 l_ρ 会连续 2 次相等，即 0，0，1，1，2，2，…，这种趋势增加。同时迭代次数 ρ 是从 1 开始到 $2t$ 结束差值为 1 的数列。$\rho-l_\rho$ 的值是随着迭代次数的增大而增大。

所以当所有的 d_μ 不为 0 时，寻找的第 ρ 次迭代，是第 $\mu-1$ 次，即 $\mu-\rho$ 为 1。这就是改进的搜索算法。此时得到改进的公式为

$$\sigma^{(\mu+1)}(x)=\sigma^{(\mu)}(x)-d_\mu d_{\mu-1}^{-1}x\sigma^{(\mu-1)}(x) \tag{14}$$

在改进搜索算法的基础上，当碰到某次 d_μ 为 0 时，找到 μ 次前最近的 $d_{\mu'}$ 不为 0 的第 μ' 次，当求解新的 $\sigma(x)$ 时，比较 $\mu'-d_{\mu'}$ 和 $\mu-1-d_{\mu-1}$ 大小，并保存下最大的值，只需比较一次。原来的 BM 算法，其搜索的次数需要 $t(1+2t)$，复杂度为 **O**(n²)。而改进的算法，搜索的次数需要 $2t$，复杂度为 **O**(n)，下降了一个数量级。

3 求解错误位置

对于 RS(n, k)，使得式(11)为 0 的值的倒数就是错误位置。试探的值就有 n 个，即 $\alpha^0,\alpha^1,\alpha^2,\cdots,\alpha^{n-2},\alpha^{n-1}$。为了方便表示，用 α^d 表示 $\alpha^0,\alpha^1,\alpha^2,\cdots,\alpha^{n-2},\alpha^{n-1}$ 这 n 个值，其中 $0\le d\le n-1$。把 α^d 代入到式(11)中，有

$$1+\sigma_1\alpha^d+\sigma_2(\alpha^d)^2\cdots+\sigma_t(\alpha^d)^t \tag{15}$$

观察式(15)，可以用循环迭代的方法求解，也就是变为

$$(\cdots(\sigma_t\alpha^d+\sigma_{t-1})\alpha^d+\cdots+\sigma_1\cdots)\alpha^d+1 \tag{16}$$

4　求错误值

4.1　通用求解错误值算法

在文献[7]介绍的求错误值算法基础上，推导了适合多种标准的求解错误值公式。所有伴随式的值都是确定的实整数值，可以写成多项式的形式：

$$s(x)=1+s_1x+\cdots+s_{2t}x^{2t}=1+\sum_{i=1}^{2t}s_ix^i \tag{17}$$

结合式(10)，$s(x)$还能表示为

$$s(x)=1+\sum_{i=1}^{2t}\sum_{k=1}^{t}Y_k(\alpha^{C*(b+i)})^{jk}x^i \tag{18}$$

其中：Y_k 为错误值，jk 作为一个整体代表错误位置，jk 的取值是 $j1$, $j2$, …, jt。$(\alpha^{C*(b+i)})^{jk}$ 这一部分过于复杂可以改写为 $\alpha^{jk(C*b+C*i)}$，也就是 $\alpha^{jk(C*b)}\alpha^{(jk*C)i}$。经过求错误位置后，$jk$ 的值是知道的，所以对于 $\alpha^{jk(C*b)}\alpha^{(jk*C)i}$ 这个式子，第一项是个常数，于是 $s(x)$可以表示为

$$\begin{aligned}s(x)&=1+\alpha^{jk(C*b)}\sum_{k=1}^{t}Y_k\sum_{i=1}^{2t}(\alpha^{(jk*C)}x)^i\\&=1+\alpha^{jk(C*b)}\sum_{k=1}^{t}Y_k\frac{\alpha^{(jk*C)}x\left(1-\left(\alpha^{(jk*C)}x\right)^{2t}\right)}{1-\alpha^{(jk*C)}x}\\&=1+\alpha^{jk(C*b)}\sum_{k=1}^{t}Y_k\frac{\alpha^{(jk*C)}x-\left(\alpha^{(jk*C)}x\right)^{2t+1}}{1-\alpha^{(jk*C)}x}\end{aligned} \tag{19}$$

定义一个错误值多项式 $w(x)$，且 $w(x)=s(x)\sigma(x)$。$s(x)$最高次为 $2t$，$\sigma(x)$的最高次为 t，二者相乘会出现 $3t$ 次，需要对 $w(x)$做 $2t$+1 求余运算，有

$$\begin{aligned}w(x)&=1+\left[\alpha^{jk(C*b)}\sum_{k=1}^{t}Y_k\frac{\alpha^{(jk*C)}x}{1-\alpha^{(jk*C)}x}\right]\sigma(x)\\&=\sigma(x)+\alpha^{jk(C*b)}\sum_{k=1}^{t}Y_k\frac{\alpha^{(jk*C)}x}{1-\alpha^{(jk*C)}x}\sigma(x)\\&=\sigma(x)+\left[\alpha^{jm(C*b)}Y_m\frac{\alpha^{(jm*C)}x}{1-\alpha^{(jm*C)}x}+\right.\\&\left.\alpha^{jk(C*b)}\sum_{\substack{k=1\\k\neq m}}^{t}Y_k\frac{\alpha^{(jk*C)}x}{1-\alpha^{(jk*C)}x}\right]\sigma(x)\\&=\sigma(x)+\alpha^{jm(C*b)}Y_m\frac{\alpha^{(jm*C)}x}{1-\alpha^{(jm*C)}x}\prod_{i=1}^{t}(1-\alpha^{C*ji}x)+\\&\alpha^{jk(C*b)}\sum_{\substack{k=1\\k\neq m}}^{t}Y_k\frac{\alpha^{(jk*C)}x}{1-\alpha^{(jk*C)}x}\prod_{i=1}^{t}(1-\alpha^{C*ji}x)\\&=\sigma(x)+\alpha^{jm(C*b)}Y_m\alpha^{(jm*C)}x\prod_{\substack{i=1\\i\neq m}}^{t}(1-\alpha^{C*ji}x)+\\&\alpha^{jk(C*b)}\sum_{\substack{k=1\\k\neq m}}^{t}Y_k\frac{\alpha^{(jk*C)}x}{1-\alpha^{(jk*C)}x}\prod_{i=1}^{t}(1-\alpha^{C*ji}x)\end{aligned} \tag{20}$$

其中：$1\le m\le t$，$1\le k\le t$，且 $k\neq m$。

错误位置 jm 是已知的。把 α^{-jm*C} 代入到式(20)中，由于 α^{-jm*C} 是 $\sigma(x)$ 的一个根，所以 $\sigma(x)$ 这一项为 0。式(20)的最后一项中的 $\prod_{i=1}^{t}(1-\alpha^{C*ji}x)$，是 $\sigma(x)$ 的另一种表达形式，所以最后一项也为 0。式(20)整理后，有

$$w(\alpha^{-jm*C})=\alpha^{jm(C*b)}Y_m\prod_{\substack{i=1\\i\neq m}}^{t}(1-\alpha^{C*ji}\alpha^{-jm*C}) \tag{21}$$

观察发现 $\prod_{\substack{i=1\\i\neq m}}^{t}(1-\alpha^{C*ji}\alpha^{-jm*C})$ 比 $\prod_{i=1}^{t}(1-\alpha^{C*ji}x)$ 少了一个幂次，尝试对 $\sigma(x)$ 求一阶导数，有

$$\begin{aligned}\sigma'(x)&=(1-\alpha^{C*j1}x)'(1-\alpha^{C*j2}x)\cdots(1-\alpha^{C*jt}x)+\\&(1-\alpha^{C*j1}x)(1-\alpha^{C*j2}x)'\cdots(1-\alpha^{C*jt}x)+\cdots+\\&(1-\alpha^{C*j1}x)(1-\alpha^{C*j2}x)\cdots(1-\alpha^{C*jt}x)'\\&=-\sum_{i=1}^{t}\alpha^{C*ji}\prod_{\substack{k=1\\k\neq i}}^{t}(1-\alpha^{C*jk}x)\end{aligned} \tag{22}$$

把 α^{-jm*C} 代入到式(22)中有

$$\sigma'(\alpha^{-jm*C})=-\alpha^{C*jm}\prod_{\substack{k=1\\k\neq m}}^{v}(1-\alpha^{C*jk}\alpha^{-jm*C}) \tag{23}$$

结合式(21)有

$$w(\alpha^{-jm*C})=\alpha^{jm(C*b)}Y_m\frac{\sigma'(\alpha^{-jm*C})}{-\alpha^{C*jm}} \tag{24}$$

即

$$Y_m=\frac{-\alpha^{C*jm}w(\alpha^{-jm*C})}{\alpha^{jm(C*b)}\sigma'(\alpha^{-jm*C})}=\frac{-w(\alpha^{-jm*C})}{\alpha^{jmb}\sigma'(\alpha^{-jm*C})}=\frac{-\alpha^{-jmb}w(\alpha^{-jm*C})}{\sigma'(\alpha^{-jm*C})} \tag{25}$$

Y_m 就是需要求的错误值。当 $s(x)$和 $\sigma(x)$ 求出来之后，$w(x)$ 和 $\sigma'(x)$ 都是已知的。RS 码的运算是在伽罗华域上进行，于是进一步把式(25)化简为

$$Y_m=\frac{w(\alpha^{-jm*C}\alpha^{-jmb})}{\sigma'(\alpha^{-jm*C})}=\frac{w(\alpha^{-jm(C+b)})}{\sigma'(\alpha^{-jm*C})}=\frac{w(\alpha^{n-jm(C+b)})}{\sigma'(\alpha^{n-jm*C})} \tag{26}$$

就是本文推导的适合多种参数的求解错误值公式。

4.2　求值公式的优化

式(26)中涉及到的 $w(x)$是一个最高次为 t，由 t 个因式组成的多项式，即 $(1-Y_1)(1-Y_2)(1-Y_3)\cdots(1-Y_t)$，其中 $1\le j\le t$。所以求解 $w(x)$时，不需要把 $s(x)$和 $\sigma(x)$ 乘加 $2t$ 次。参与求解的 $s(x)$只需常数项和小于次数 t+1 的系数参与，而 $\sigma(x)$ 的所有项都参与，这样会省去一半运算。于是有 $w(x)=1+w_1x+w_2x^2+\cdots+w_tx^t$，且

$$\begin{aligned}w(x)=1&+(s_1+\sigma_1)x+\\&(s_2+s_1\sigma_1+\sigma_2)x^2+\ldots+\\&(s_t+s_1\sigma_{t-1}+\cdots+s_{t-1}\sigma_1+\sigma_t)x^t\end{aligned} \tag{27}$$

5　卫星通信中 CCSDS 的 RS 译码仿真测试

测试的 RS 码是 CCSDS 标准下的 RS(255, 223)[12]。输入的信息数据是 1, 2, 3, …,221, 222, 223 这 223 个信息符号。RS 编码后产生的 32 个校验信息分别是[223, 143, 243, 66, 0, 177, 182, 232, 176, 79, 114, 129, 85, 7, 223, 153, 129, 150, 94, 238, 241, 200, 6, 100, 229, 108, 173, 61, 98, 107, 173, 240]。那么编码后的 255 个数据为[1, 2, 3,…, 222, 223, 223, 143, 243, 66, 0, 177, 182, 232, 176, 79, 114, 129, 85, 7, 223, 153, 129, 150, 94, 238, 241, 200, 6, 100, 229, 108, 173, 61, 98, 107, 173, 240]。那么这 255 个数据从左到右，分别对应于发送信息多项式中的 $c_{n-1},c_{n-2},\cdots,c_1,c_0$。为了验证本文提出的 RS 译码算法那的准确性，令编码后的 32 个校验数据的前 16 个数据为 1 到 16，也就是将编码后的数据变为 [1, 2, 3, … , 222, 223, 1, 2, 3, … , 14, 15, 16 , 129, 150, 94, 238, 241, 200, 6, 100, 229, 108, 173, 61, 98, 107, 173, 240]，对应接收多项式中的 $r_{n-1},r_{n-2},\cdots,r_1,r_0$。计算后得到的伴随式为[80, 194, 209, 59, 138, 42, 166, 186, 62, 249, 35, 8, 226, 71, 92, 184, 126, 58, 88, 203, 99, 42, 131, 225, 17, 110, 48, 216, 73, 7, 102, 241]。

应用改进搜索算法，产生的错误位置多项式的系数为[68, 40, 156, 53, 121, 91, 24, 126, 158, 88, 97, 123, 31, 134, 146, 251]，一共 16 个。产生的 32 个差值 d_μ 为 [80, 152, 138, 172, 10, 243, 237, 13, 158, 140, 78, 176, 247, 19, 23, 206, 167, 130, 179, 70, 171, 205, 79, 193, 220, 31, 197, 55, 172, 205, 224, 178]，一共 32 个。每次运行产生的 $\mu-l_\mu$ 为[0, 0, 1, 1, 2, 2, 3,

3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15]，一共 32 个，这些值也正好验证了改进 BM 搜索算法的正确性。接下来求得错误位置为[224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239]，相应的错误值为[222, 141, 240, 70, 5, 183, 177, 224, 185, 69, 121, 141, 88, 55, 208, 137]。在伽罗华域里面，把接收信息中错误位置的错误值和 RS 译码求出的错误值进行相加，就能得到正确的结果。也就是把[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]和[222, 141, 240, 70, 5, 183, 177, 224, 185, 69, 121, 141, 88, 55, 208, 137]对应相加，得到[223, 143, 243, 66, 0, 177, 182, 232, 176, 79, 114, 129, 85, 7, 223, 153]。修正后的 16 个值和编码后 32 个校验值的前 16 个值是相等的。既原先加入的 16 个错误，得到了纠正。

接下来，调制方式采用 BPSK，噪声为高斯白噪声，在 MATLAB 环境下进行误比特率测试实验。RS 码采用 CCSDS 的 RS(255, 223)码。一个符号为 8 bit，码长 255 个符号，信息长度为 223 个符号。编码时涉及到的系数为[1, 91, 127, 86, 16, 30, 13, 235, 97, 165, 8, 42, 54, 86, 171, 32, 113, 32, 171, 86, 54, 42, 8, 165, 97, 235, 13, 30, 16, 86, 127, 91, 1]。然后根据改进的译码算法，运行百万次测试后得到误比特率曲线，并与理论未编码的 BPSK 曲线进行对比，如图 2 所示。

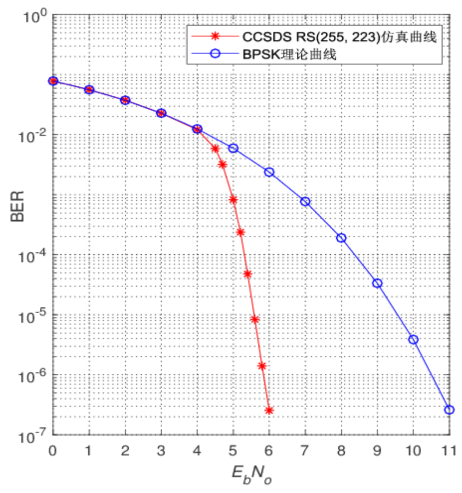


图 2　RS(255, 223)误比特率曲线

Fig. 2　The BER curve of RS(255, 223)

观察图 2 可以发现，信噪比为 0~4 dB，RS(255,223)的曲线和理论值基本一致。但随着信噪比的不断增大，RS(255,223)的曲线开始陡峭，误比特率急速下降。在误比特率为 10^{-7} 时，RS(255,223)比未编码的 BPSK 多获得 5 dB 的增益。从而验证了改进译码算法的正确性。

6　结束语

本文对 RS 译码算法进行了深入分析，对求解伴随式、错误位置多项式、错误位置和错误值都进行了研究。其中，通常采用 BM 算法求解错误位置多项式。BM 算法中关键步骤是搜索某一步迭代，本文分析了一种改进的搜索算法，使得搜索复杂度降低了一个数量级；对求解错误位置也进行了优化，结合传统算法，推导了一个更通用的公式。以 CCSDS 标准下 RS(255,223)为研究对象，采用改进的译码算法求解错误位置和求解错误值，进行了具体的数值实验测试和误比特率测试。测试结果表明，采用改进的译码算法可以快速正确译码，同时验证了改进算法的正确性。进一步的工作包括: a)对伴随式模块进行优化研究，根据伴随式的值判断错误发生的情况; b)在硬件电路上实现 RS 高速并行译码器和低功耗串行译码器。

参考文献：

[1] Wang Tian, Zhou Jiyuan, Liu Anfeng, *et al.* Fog-based computing and storage offloading for data synchronization in IoT [J]. IEEE Internet of Things Journal, 2018, DOI: 10. 1109//JIOT. 2018. 2875915.

[2] Demirkan I, Silvus G. Multilevel Reed–Solomon codes for PMR channels [J]. IEEE Trans on Magnetics, 2016, 52(2): 1-8.

[3] Li Yong, Huang Xiang, He Jiguang, *et al.* On soft-information-based error and erasure decoding of Reed-Solomon codes in burst rayleigh fading channels [J]. IEEE Trans on Communications, 2018, DOI: 10. 1109//TCOMM. 2018. 2872033.

[4] Dissanayake M B, Deng Y, Nallanathan A, *et al.* Reed Solomon codes for molecular communication with full absorption receiver [J]. IEEE Communications Letters, 2017, 21(6): 1-4.

[5] Chen Ming, Xiao Xin, Li Xinying, *et al.* Improved BER performance of real-time DDO-OFDM systems using interleaved Reed–Solomon codes [J]. IEEE Photonics Technology Letters, 2016, 28(9): 1014-1017.

[6] Borujeny R R, Ardakani I M. A new class of rateless codes based on Reed–Solomon codes [J]. IEEE Trans on Communications, 2016, 64(1): 49-58.

[7] 张宗橙. 纠错编码原理和应用[M].北京:电子工业出版社,2003: 120-130.

[8] Moon T K. Error correction coding: mathematical methods and algorithms [M]. Hoboken: Wiley, 2005: 235-280.

[9] Massey J L. Shift-register synthesis and BCH decoding [J]. IEEE Trans on Information Theory, 1969, 15(1): 122-127.

[10] Park J I, Lee K, Choi C S, *et al.* High-speed low-complexity Reed-Solomon decoder using pipelined Berlekamp-Massey algorithm and its folded architecture [J]. Journal of Semiconductor Technology & Science, 2010, 10(3): 193-202.

[11] Jorge C M, Patrick G F. Essentials of error-control coding [M]. Hoboken: Wiley, 2006: 100-153.

[12] CCSDS Secretariat. CCSDS 131. 0-B-3 Blue Book, TM synchronization and channel coding [S]. Washington DC: Management Council of CCSDS, 2017.